

su 1

SISTEMAS DE ALMACENAMIENTO DE LA INFORMACIÓN

Índice

Gestión de datos.....	3
Componentes de un sistema informático	4
Tipos de sistemas informáticos.....	4
Tipos de sistemas de información de gestión de datos.....	5
• Sistemas de información orientados al proceso: FICHEROS	5
• Sistemas de información orientados a los datos: BASES DE DATOS	6
Sistemas de archivos	7
Tipos de ficheros según su estructura física.....	7
Tipos de archivos según su uso	11
Sistemas de Bases de datos	12
Componentes de un sistema de base de datos.....	13
Arquitectura de los sistemas de bases de datos	17
Estructuras operacionales.....	19
Sistemas Gestores de Bases de datos	19
Funciones de los SGBD	21
Ventajas	24
Inconvenientes	24
Lenguajes de los SGBD	24
Lenguaje de definición de datos (DDL)	25
Lenguaje de manejo de datos (DML)	25
Lenguaje de control de datos (DCL)	26
Clasificación de los sistemas gestores de bases de datos	26
Según el modelo lógico utilizado	26
Según el número de usuarios a los que presta servicio.....	27
Según la distribución física de la base de datos	27

Gestión de datos

En el mundo actual existe una demanda cada vez mayor de datos y, por tanto, mayor necesidad de gestionarlos. Esta demanda siempre ha sido patente en empresas y sociedades, pero en estos años la demanda todavía se ha disparado más debido al acceso multitudinario a las redes integradas en Internet ya la aparición de pequeños dispositivos (móviles y PDA) que también requieren esa información.

En informática se conoce como dato a cualquier elemento informativo que tenga relevancia para un usuario. Desde su nacimiento, la informática se ha encargado de proporcionar herramientas que faciliten la manipulación de los datos.

Los sistemas de información actuales se basan en bases de datos y sistemas de bases de datos, que se han convertido en elementos imprescindibles de la vida cotidiana de la sociedad moderna.

Todos los días, la mayoría de nosotros nos encontramos con actividades que requieren algún tipo de interacción con una base de datos (ingreso en un banco, reserva de una entrada para el teatro, solicitud de una suscripción a una revista, compra de productos, ...). Estas interacciones son ejemplos de lo que se llama aplicaciones tradicionales de bases de datos.

Una base de datos será, por tanto, una colección de datos relacionados. Por datos queremos decir hechos conocidos que pueden registrarse y que tienen un significado implícito. Una agenda con los nombres, fotografías y teléfonos de un conjunto de personas conocidas es una base de datos, ya que es una colección de datos relacionadas con un significado implícito.

Actualmente, debido a la gran cantidad de información con la que se cuenta, aparece el concepto de Big Data. Denominamos así a la gestión y análisis de enormes volúmenes de datos que no pueden ser tratados de forma convencional, ya que superan los límites y capacidades de las herramientas de software habitualmente utilizadas para la captura, gestión y procesamiento de datos.

Este concepto engloba infraestructuras, tecnologías y servicios que han sido creados para dar solución al procesamiento de enormes conjuntos de datos estructuradas, no estructuradas o semi-estructuradas (mensajes en redes sociales, señales de móvil, archivos de audio, sensores, imágenes digitales, datos de formularios, emails, datos de encuestas, logs etc.) que pueden provenir de sensores, micrófonos, cámaras, escáneres médicos, imágenes, ...

El objetivo de Big Data, al igual que los sistemas analíticos convencionales, es convertir el dato en información que facilita la toma de decisiones, incluso en tiempo real.

Sin embargo, más que una cuestión de tamaño, es una oportunidad de negocio. Las empresas ya están utilizando Big Data para entender el perfil, las necesidades y el sentir de sus clientes con respecto a los productos y/o servicios vendidos. Esto adquiere especial relevancia ya que permite adecuar la forma en la que interactúa la empresa con sus clientes y en cómo les prestan servicio.

Sistemas de información.

Los sistemas que aglutinan los elementos que intervienen para gestionar la información que manejan cualquier entidad, empresa o institución es lo que se conoce como Sistemas de Información .

Un sistema de información es un sistema que almacena y gestiona la información relevante de una empresa/institución/entidad.

Un sistema de información genérico está formado por los siguientes elementos:

- Recursos físicos. Carpetas, documentos, equipamiento, discos,...
- Recursos humanos. Personal que maneja la información.
- Reglas. Normas que debe cumplir la información para que sea manejada (formato de la información, modelo para los documentos...), es decir, las normas, métodos y protocolos determinados por la planificación de la empresa.

Cuando parte o toda la gestión de un sistema de información se realiza con ordenadores, se habla de sistema de información basado en ordenadores o sistema informático. En este caso en las reglas de la empresa se añaden las normas determinadas por el sistema operativo de los ordenadores y otro software instalado en ellos.

Componentes de un sistema informático

- Datos. Se trata de la información relevante que almacena y gestiona el sistema de información. Esta información suele estar almacenada en bases de datos que cumplen con los requerimientos o normas de la empresa.
- Hardware. Equipamiento físico que se utiliza para manejar los datos.
- Software. Aplicaciones que permiten el adecuado funcionamiento del sistema.
- Recursos humanos. Personal que maneja el sistema de información.

Tipo de sistemas informáticos

Según el propósito, los sistemas de información se pueden clasificar en:

- Transaccionales. Se ocupan de la automatización de las operaciones y transacciones

que se realizan en la empresa, por ejemplo, las actividades que realizan los empleados de forma cotidiana (fichar la hora de entrada del trabajo, firmar, ...)

- **De Gestión.** Se ocupan de los datos que se manejan en la empresa y su almacenamiento. Por ejemplo controlan cómo se almacenan los datos en las bases de datos, cómo se recupera esa información, cómo se obtienen listados, informes, etc.
- **Ofimáticos.** Se utilizan para realizar de forma mecanizada las múltiples tareas de una oficina: automatización de correspondencia, creación de documentos para imprimir, mantenimiento de datos que requieren continuos cálculos, etc.
- **De apoyo a la decisión.** Su misión es ayudar a los directivos y personal con responsabilidad dentro de la empresa en la toma de decisiones estratégicas. Estos sistemas están emparentados con las técnicas de inteligencia empresarial y los sistemas expertos y son cada vez más comunes e importantes dentro de las empresas.
- **Inteligentes.** Son aplicaciones que reúnen características y comportamientos similares al de la inteligencia humana o animal. Un sistema inteligente completo incluye "sentidos" que le permiten recibir información de su entorno. Puede actuar, y tiene una memoria para archivar el resultado de sus acciones. Tiene un objetivo y, inspeccionando su memoria, puede aprender de su experiencia. Aprende cómo conseguir mejorar su rendimiento y eficiencia.

Tipo de sistemas de información de gestión de datos

Los sistemas de información pueden diferenciarse en orientados al proceso (sistema clásico de archivos) u orientados a los datos (sistema de bases de datos).

El uso de uno u otro sistema depende de la fecha en la que el sistema informático fue implantado en la empresa. En los años 60/70 se usaban archivos de datos a los que se accedía a través de lenguajes de programación como Cobol, C, Basic, .. A mediados de 70 se crean los primeros gestores de bases de datos que evolucionan hasta los conocidos relacionales SQL Server, ORACLE, MySQL... En la actualidad, la gran cantidad de datos a procesar requiere de bases de datos avanzadas que aportan una mezcla entre la idea de orientación a proceso y orientación a datos: las denominadas NoSQL como MongoDB.

• Sistemas de información orientados al proceso: ARCHIVOS

En estos sistemas de información se crean diversas aplicaciones para gestionar distintos aspectos del sistema. Cada aplicación realiza unas determinadas funcionalidades.

Los datos de estas aplicaciones se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador (en archivos binarios, de texto, hojas de cálculo,...).

En estos sistemas, cada programa almacena y utiliza sus propios datos de forma algo caótica. La única ventaja que conlleva esto es que los procesos son independientes, por lo que la modificación de uno no afecta al resto. Pero tiene grandes inconvenientes:

- Coste de almacenamiento elevado. Al almacenarse varias veces el mismo dato en distintas aplicaciones, se requiere más espacio en los discos.
- Datos redundantes. Ya que se repiten continuamente.
- Probabilidad alta de inconsistencia en los datos. Puesto que un proceso cambia sus datos y no el resto. Por lo que el mismo dato puede tener valores diferentes según qué aplicación acceda a él.
- Dificil modificación en los datos Debido a la probabilidad de inconsistencia, que ocurre cuando se produce una pérdida o existe incoherencia de datos. Para que ésta no exista, cada modificación debe repetirse en todas las copias del dato (algo que normalmente es imposible).
- Tiempos de procesamiento elevados. Al no poder optimizar el espacio de almacenamiento.

• Sistemas de información orientados a los datos: BASES DE DATOS

En este tipo de sistemas, los datos se centralizan en una base de datos común a todas las aplicaciones. Estos serán los sistemas que estudiaremos en este curso.

En esos sistemas los datos se almacenan en una única estructura lógica que es utilizable por las aplicaciones. A través de esa estructura se accede a los datos que son comunes a todas las aplicaciones.

Ventajas

- Independencia de los datos y programas y procesos. Esto permite modificar los datos sin modificar el código de las aplicaciones.
- Menor redundancia. No hace falta tanta repetición de datos.
- Integridad de los datos. Mayor dificultad de perder los datos o realizar incoherencias con ellos.
- Mayor seguridad en los datos. Al limitar el acceso a ciertos usuarios.
- Datos más documentados. Gracias a los metadatos que permiten describir la información de la base de datos.
- Acceso a los datos más eficiente. La organización de los datos produce un resultado más óptimo en rendimiento.
- Menor espacio de almacenamiento. Gracias a una mejor estructuración de datos.

Inconvenientes

- Instalación costosa. El control y administración de bases de datos requiere de un software y hardware poderoso
- Requiere personal cualificado. Debido a la dificultad de manejo de este tipo de sistemas.
- Implantación larga y difícil. Debido a los puntos anteriores. La adaptación del personal es mucho más complicada y lleva bastante tiempo.

Sistemas de archivos

Un sistema de archivos es un conjunto de programas que prestan servicio a los usuarios finales de forma que cada programa define y maneja sus propios datos.

Los sistemas de archivos surgieron al tratar de informatizar el manejo de los archivadores manuales a fin de proporcionar un acceso más eficiente a los datos.

Los archivos o archivos son la herramienta fundamental de trabajo en un ordenador aún hoy en día. Estos archivos se almacenan en componentes de almacenamiento permanente.

Un archivo o archivo es una secuencia de caracteres o de datos binarios que organiza información relacionada a un mismo aspecto. Por lo general, sobre los archivos se pueden realizar las siguientes operaciones:

- Abrir (open). Prepara el archivo para su proceso.
- Cerrar (close). Cierra el archivo impidiendo su proceso inmediato.
- Leer (read). Obtiene información del archivo.
- Escribir (write). Graba información en el archivo.
- Posicionarse (seek). Coloca el puntero de lectura en una posición concreta del mismo (no se puede realizar en todos los tipos de archivos).

Cuando en los archivos almacenan datos, se dice que constan de registros.

Cada registro contiene datos relativos a un mismo elemento u objeto. Por ejemplo, en un archivo de personas cada registro contiene datos de una persona. Si el archivo contiene datos de 1000 personas, constará de 1000 registros. Cada característica de ese registro se denomina campo, por ejemplo, la fecha de nacimiento de una persona.

Tipo de ficheros según su estructura física

• Archivos secuenciales

En estos archivos, los datos se organizan secuencialmente en el orden en el que fueron grabados. Para leer algún registro de datos es necesario leer todos los anteriores. Sus características son:

- Rápidos para obtener registros contiguos.
- No hay vacíos en el archivo al grabarse los datos seguidos por lo que los datos son más compacto.
- Consultas muy lentas al tener que leer todos los datos anteriores al dato que queremos leer
- Algoritmos de lectura y escritura más complejos
- No se pueden eliminar registros del archivo (se pueden marcar de manera especial para que no sean tenidos en cuenta, pero no se pueden borrar)
- La ordenación de los datos requiere volver a crearlo de nuevo

- Archivos de acceso directo o aleatorio

Los registros se graban en posiciones fijas dentro del archivo, habitualmente una posición numérica relacionada con algún campo de su contenido que se ha deducido mediante un algoritmo. Por tanto, se puede acceder a una posición concreta del archivo con saber la posición (normalmente en bytes) del registro a tratar.



Sus características son:

- Acceso rápido al no tener que leer los datos anteriores
- La modificación de datos es más sencilla
- Permiten acceso secuencial
- Permiten leer y escribir a la vez
- Aptas para organizaciones relativas directas, en las que la clave del registro se relaciona con su posición en el archivo
- No se pueden borrar datos (sí que marcar para borrado, pero se generarán vacíos)
- Las consultas sobre multitud de registros son más lentas que en el caso anterior

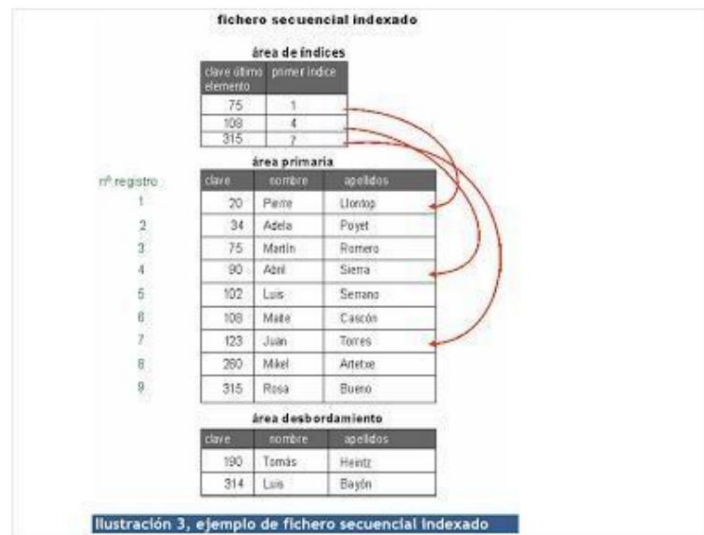
- Archivos secuenciales encadenados

Son archivos secuenciales gestionados mediante punteros, datos especiales que contienen la dirección de cada registro del archivo. Cada registro posee ese puntero que indica la dirección del siguiente registro y que se puede modificar en cualquier momento. El puntero permite recorrer los datos en un orden concreto.



Cuando aparece un nuevo registro, se añade al final del archivo, pero los punteros se reordenan para que se mantenga el orden. Sus características son:

- El archivo mantiene el orden en el que se añadieron los registros y un segundo orden sobre la base de una clave
 - La ordenación no requiere reorganizar todo el archivo, sino sólo modificar los punteros
 - Las mismas ventajas que el acceso secuencial
 - En este caso sí que se borran los registros y al reorganizar, se perderán definitivamente.
 - No se borran los registros, sino que se marcan para ser ignorados. Por lo que se desperdicia espacio
 - Añadir registros o modificar las claves requieren recalcular los punteros
- **Archivos secuenciales indexados**



Se utilizan dos archivos, uno posee los registros almacenados de manera secuencial, pero que permite su acceso aleatorio (de datos o principal). El otro posee una tabla con punteros en la posición ordenada

de los registros (índice). Ese segundo archivo una tabla con la ordenación deseada para los registros y la posición que ocupan en el archivo.

El archivo de índices posee unas cuantas entradas sólo en las que se indica la posición de ciertos valores claves en el archivo (cada 10, 15, 20,... registros del archivo principal se añade una entrada en el de índices).

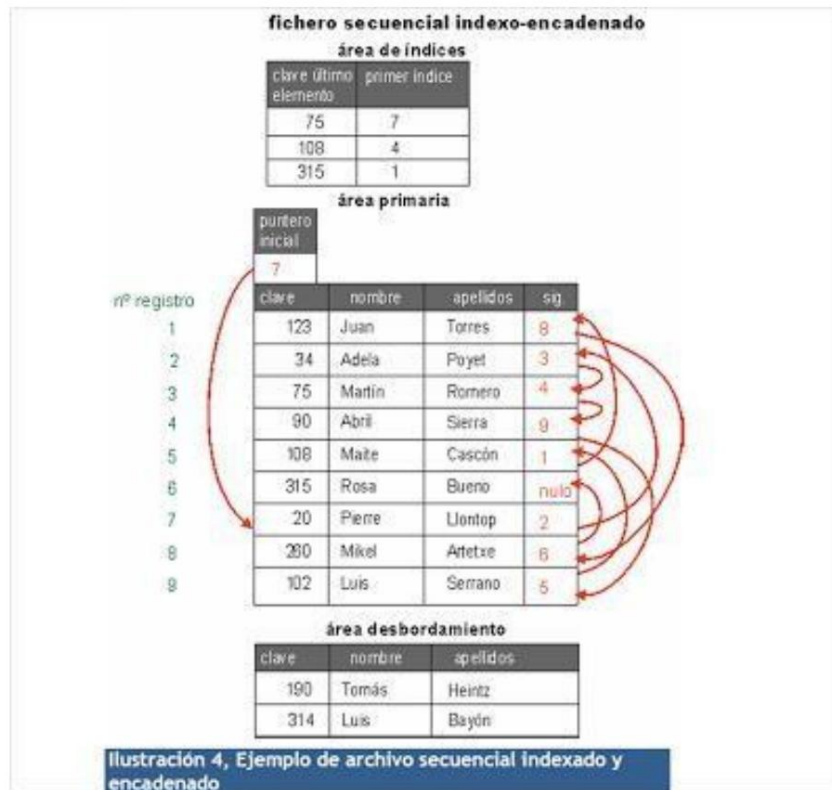
El archivo principal debe estar siempre ordenado y así cuando se busca un registro, se busca su valor clave en la tabla de índices, que poseerá la posición del registro buscado. Desde esa posición se busca secuencialmente el registro hasta encontrarlo.

Existe un archivo llamado de desbordamiento u overflow en el que se colocan los nuevos registros que se van añadiendo (para no tener que ordenar el archivo principal cada vez que se añade un nuevo registro) este archivo está desordenado. Se utiliza sólo si se busca un registro y no se encuentra en el archivo principal. En ese caso se recorre todo el archivo de overflow hasta encontrarlo. Para no tener demasiados archivos en overflow (lo que restaría velocidad), cada cierto tiempo se reorganiza el archivo principal.

- El archivo está siempre ordenado sobre la base de una clave
- La búsqueda de datos es rapidísima
- Permite la lectura secuencial (que además será en el orden de la clave)
- El borrado de registros es posible (aunque más problemático que en el caso anterior)
- Para un uso óptimo es necesario reorganizar el archivo principal y esta operación es muy costosa ya que es necesario reescribir de nuevo y de forma ordenada todo el archivo.
- La adición de registros requiere más tiempo que en los casos anteriores al tener que reordenar los índices.

• Archivos indexado-encadenados

Variant encadenada del cas anterior.



Posee las mismas ventajas que los archivos secuenciales indexados, además de una mayor rapidez al reorganizar el archivo (sólo se modifican los punteros)

- Requieren compactar los datos a menudo para reorganizar índices y quitar el archivo de desbordamiento.

Tipo de archivos según su uso

- **Maestros.** Archivos que se cambian muy poco a menudo. Son los principales en los sistemas de información. Contienen los datos fundamentales de una organización o empresa.
- **Constantes.** Contienen información que apenas varía a lo largo del tiempo.
- **Históricos.** Contienen los datos de los archivos maestros pasado un determinado tiempo.
- **De movimiento.** Almacenan cambios a realizar en los archivos maestros. Se eliminan después de realizar esos cambios.
- **De maniobra.** Archivos auxiliares utilizados por el software que gestiona las datos. Son destruidos cuando finalizan las aplicaciones.

Sistemas de Bases de datos

Un Sistema de Bases de datos es un sistema basado en ordenadores, cuyo propósito general es registrar y mantener datos mediante un sistema gestor de BD.

Permite a los usuarios y aplicaciones la consulta y manipulación de estos datos almacenadas en bases de datos

Como se ha comentado anteriormente, cuando los datos de un sistema de información se almacenan en una única estructura, se llama base de datos. Se presentan a continuación dos definiciones:

Una base de datos es una colección de datos almacenados en un soporte informático permanente de forma que sea posible obtener la relación entre los datos a través de un esquema conceptual que oculte la física real de los datos.

Una base de datos es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real. Los datos, que deben ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ésta, y su definición y descripción deben ser únicas estando almacenados junto a éstos. Por último, los tratamientos que sufran estos datos deberán conservar la integridad y seguridad de estos..

El objetivo es que las aplicaciones puedan acceder a los datos sin necesidad de conocer exactamente cómo están almacenados los datos. Esto se logra con un esquema conocido como esquema conceptual. Cualquiera desarrollador que conozca ese esquema puede acceder a los datos desde cualquier aplicación.

Las bases de datos están compuestas (como ya se han comentado), de datos y de metadatos. Los metadatos son datos (valga la redundancia) que sirven para especificar la estructura de la base de datos; por ejemplo qué tipo de datos se almacenan (si son texto o números o fechas...), qué nombre se le da a cada dato (nombre, apellidos,...), cómo están agrupados, cómo se relacionan,... De esta manera se producen dos visiones de la base de datos: lógica (las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando) y física (datos tal cual se almacenan las unidades de disco (mediante archivos directos, indexados,...)).

Componentes de un sistema de base de datos.

Los elementos de un sistema de base de datos son los mismos que los de un sistema de información. Se comentan a continuación para concretar un poco más la información que ya poseíamos:

- **Hardware.** Máquinas en las que se almacenan las bases de datos.
Se compone de los volúmenes de almacenamiento secundario (discos, cintas, ...) donde reside la BD, junto con los dispositivos asociados como unidad de control, tarjetas, memoria, ...
- **Software.** Es el sistema gestor de bases de datos, es decir, la aplicación que permite el manejo de la base de datos. Entre la BD física y los usuarios del sistema, existe un nivel de software que recibe el nombre de SGBD. Éste maneja todas las solicitudes de acceso a la BD y registra y mantiene de forma controlada los datos almacenados. Asimismo pueden existir uno o varios programas para permitir el acceso a las BD realizando consultas. Estas aplicaciones accederán a las BD usando el lenguaje de consultas SQL mediante los servicios ofrecidos por el SGBD.
- **Datos.** Incluyen los datos que se necesitan almacenar y los metadatos que son datos que sirven para describir lo que se almacena en la base de datos. Los datos almacenados en el sistema se dividen en una o más bases de datos. Una BD está integrada, ya que puede considerarse como una unificación de varios archivos de datos independientes, donde se eliminan cualquier redundancia entre éstos. Una BD es compartida, en el sentido en que partes individuales de la BD pueden compartirse entre varios usuarios distintos, de tal forma que cada uno de ellos puede tener acceso a la misma parte de la BD. Es necesario considerar que aunque dos usuarios compartan el mismo subconjunto de la BD sus percepciones o vistas de ese conjunto pueden diferir mucho a nivel de detalle. La palabra "compartida" a menudo se amplía para abarcar no sólo al comportamiento nos descrito, sino también al acceso concurrente, es decir, la posibilidad de que varios usuarios accedan a la misma BD (tal vez incluso en la misma parte) al mismo tiempo. Un SBD que admite esta forma de acceso se llama Sistema de Usuarios Múltiples.
- **Usuarios.** Personas que manipulan los datos del sistema. Hay cuatro grupos de personas que intervienen en torno a una base de datos: el administrador de la base de datos, los diseñadores de la base de datos, los programadores de aplicaciones y los usuarios.
 - El administrador de la base de datos (ABD o DBA) se encarga del diseño físico de la base de datos y de su implementación, realiza el control de la seguridad y de la concurrencia, mantiene el sistema para que

siempre se encuentre operativo y se encarga de que los usuarios y aplicaciones obtengan buenas prestaciones. El administrador debe conocer muy bien el SGBD que se esté utilizando, así como el equipo informático sobre el que esté funcionando. El DBA, en resumen, autoriza el acceso a la BD, coordina y vigila su utilización, adquiere los recursos de software y hardware necesario y es el responsable ante los problemas de violaciones de seguridad o respuesta lenta del sistema.

- Los analistas o diseñadoras de la base de datos realizan el diseño lógico de la base de datos, debiendo identificar los datos, las relaciones entre datos y restricciones sobre los datos y sus relaciones. El diseñador de la base de datos debe tener un profundo conocimiento de los datos de la empresa y también debe conocer sus reglas de negocio. Las reglas de negocio describen las principales características de los datos tal como las ve la empresa. Para obtener un buen resultado, el diseñador de la base de datos debe implicar en el desarrollo del modelo de datos a todos los usuarios de la base de datos, tan pronto como sea posible. El diseño lógico de la base de datos es independiente del SGBD concreto que se vaya a utilizar, es independiente de los programas de aplicación, de los lenguajes de programación y de cualquier otra consideración física. El diseñador, en resumen, identifica los datos que se almacenarán en la BD, elige las estructuras apropiadas, se comunica con los futuros usuarios de la BD con el fin de comprender sus necesidades y requerimientos.
- Una vez se ha diseñado e implementado la base de datos, los programadores de aplicaciones se encargan de implementar los programas de aplicación que servirán a los usuarios finales. Estos programas de aplicación son los que permiten consultar datos, insertarlos, actualizarlos y eliminarlos. Estos programas se escriben mediante lenguajes de tercera generación o de cuarta generación. El programador, en resumen, implementa las especificaciones realizadas por los analistas/diseñadores en forma de programas y después prueba, depura, documenta y mantiene estas transacciones programadas, debiendo conocer a la perfección toda la gama de capacidades del SGBD.
- Los usuarios finales son los clientes de la base de datos: la base de datos ha sido diseñada e implementada, y está siendo mantenida, para satisfacer sus requisitos en la gestión de su información.

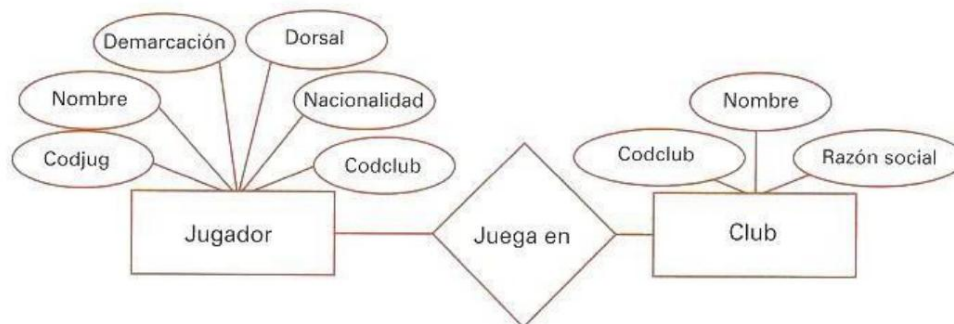
Hay que tener en cuenta que las necesidades de los usuarios son muy distintas en función del tipo de usuario que sean: a los finales les interesa la facilidad de uso, a los desarrolladores la potencia y flexibilidad de los lenguajes incorporados del sistema de bases de datos y los administradores herramientas de gestión avanzada para la base de datos.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos.

Los modelos de datos pueden clasificarse dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos. Los modelos de datos de alto nivel, o modelos conceptuales, disponen de conceptos muy próximos a la forma en que la mayoría de los usuarios percibe los datos, mientras que los modelos de datos de bajo nivel, o modelos físicos, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador. Los conceptos de modelos físicos están dirigidos al personal informático, no a los usuarios finales. Entre estos dos extremos se encuentran los modelos lógicos, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente. Los modelos lógicos ocultan

algunos detalles de cómo se almacenan los datos, pero pueden implementarse de forma directa en un ordenador.

- Los modelos conceptuales utilizan conceptos como entidades, atributos y relaciones. Una entidad representa un objeto o concepto del mundo real, como, por ejemplo, un jugador de un club deportivo. Un atributo representa alguna propiedad de interés de una entidad como, por ejemplo, el nombre o salario del empleado. Una relación describe una interacción entre dos o más entidades, por ejemplo, la relación "jugar en" entre un jugador y su club. Un ejemplo de representación del modelo conceptual sería el siguiente:

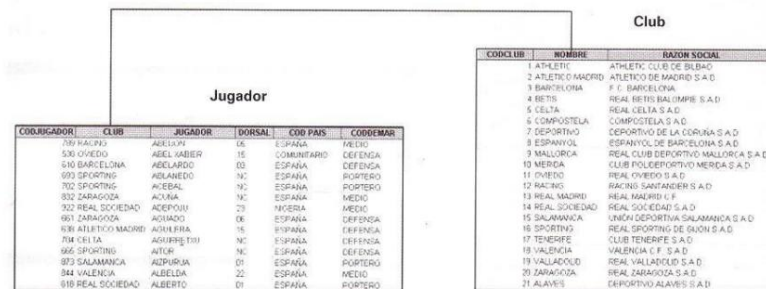


- Cada SGBD soporta un modelo lógico, siendo los más comunes el relacional, el de red y el jerárquico. Estos modelos representan los datos valiéndose de estructuras de registros, por lo que también se denominan modelos orientados a registros. Hay otra familia de modelos lógicos, son los modelos orientados a objetos, más cercanos a los modelos conceptuales.

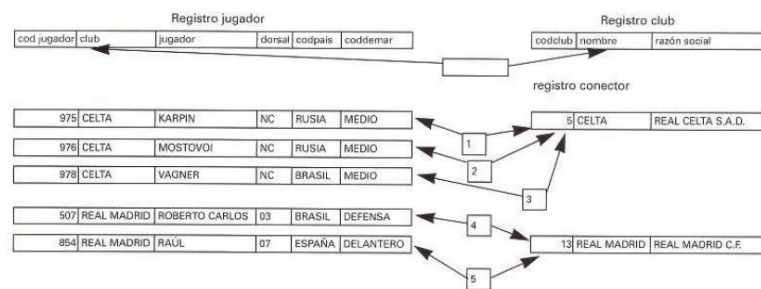
Será el modelo relacional el que será el caso de estudio durante este curso.

Las siguientes figuras muestran tres modelos lógicos del propio modelo conceptual.

Relacional



En red



Jerárquico



- Los modelos físicos describen cómo se almacenan los datos en el ordenador: el formato de los registros, la estructura de los archivos (desordenados, ordenados, etc.) y los métodos de acceso utilizados (índices, etc.).

A la descripción de una base de datos mediante un modelo de datos se le denomina **esquema de la base de datos**. Este esquema se especifica durante el diseño, y no es esperar a que se modifique a menudo. Sin embargo, los datos que se almacenan en la base de datos pueden cambiar con mucha frecuencia: se insertan datos, se actualizan, etc. Los datos que la base de datos contiene en un determinado momento se denominan **estado de la base de datos u ocurrencia de la base de datos**.

La distinción entre el esquema y el estado de la base de datos es muy importante. Cuando definimos una nueva base de datos, sólo especificamos su esquema en el SGBD. En ese momento, el estado de la base de datos es el "estado vacío", sin datos. Cuando se cargan datos por primera vez, la base de datos pasa al "estado inicial". De aquí en adelante, siempre que se realice una operación de actualización de la base de datos, se tendrá un nuevo estado. El SGBD se encarga, en parte, de garantizar que todos los estados de la base de datos sean estados válidos que satisfagan la estructura y las restricciones especificadas en el esquema. Por tanto, es muy importante que el esquema que se especifique en el SGBD sea correcto y se debe tener muchísimo cuidado al diseñarlo.

consultar siempre que sea necesario.

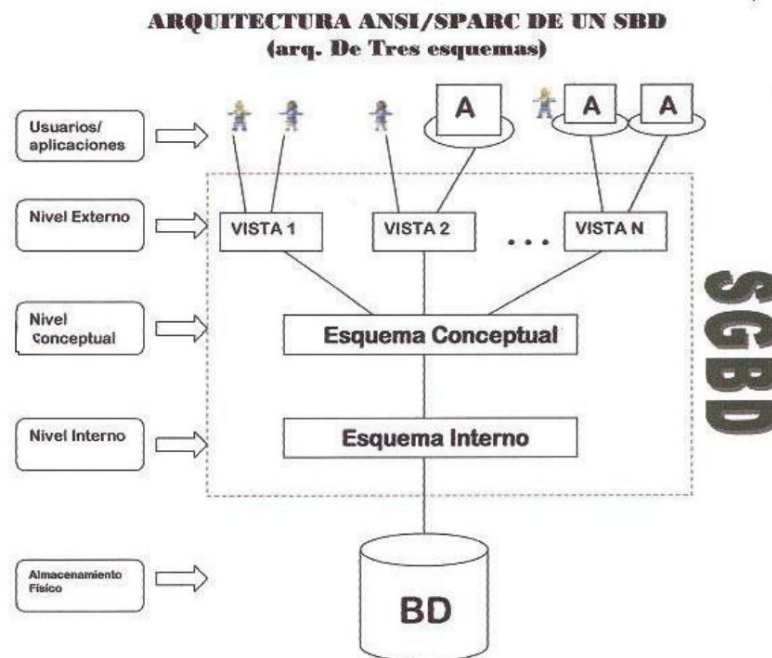
Arquitectura de los sistemas de bases de datos

En 1975, el comité ANSI-SPARC (American National Standard Institute - (Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la vez conseguir estas tres características.

El objetivo de la arquitectura de tres niveles es separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción diferentes:

- En el nivel **interno o físico** se describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
- En el nivel **conceptual** se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.
- En el nivel **externo** se describen varios esquemas externos o vistas de usuario.

Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos. En ese nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.



La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. Se pueden definir dos tipos de independencia de datos:

- La independencia lógica es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.
- La independencia física es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o externos). Por ejemplo, puede ser necesario reorganizar ciertos archivos físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

Estructuras operacionales

Actualmente casi todos los sistemas gestores de base de datos poseen también la idea operacional en la que se entiende que la base de datos se almacena en un servidor y hay una serie de clientes que pueden acceder a los datos de éste. Las posibilidades son:

- **Cliente-Servidor.** Estructura clásica, la base de datos y su SGBD están en un servidor al que acceden los clientes. El cliente posee software que permite al usuario enviar instrucciones al SGBD en el servidor y recibir los resultados de estas instrucciones. Para ello el software cliente y el servidor deben utilizar software de comunicaciones en red.
- **Cliente Multi-servidor.** Ocurre cuando los clientes acceden a datos ubicadas en más de un servidor. También se conoce esta estructura como base de datos distribuida. El cliente no sabe si los datos están en uno o más servidores, ya que el resultado es el mismo independientemente de dónde se almacenan los datos. En esta estructura existe un servidor de aplicaciones que es el que recibe las peticiones y el encargado de traducirlas a los distintos servidores de datos para obtener los resultados.
- **Cliente/Servidor Web/Servidor de datos,** el cliente se conecta a un servidor mediante un navegador web y desde las páginas del mismo ejecuta las consultas. El servidor web traduce esta consulta al servidor(o servidores) de datos.

Sistemas Gestores de Bases de datos

Un sistema gestor de bases de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener una base de datos, y proporciona acceso controlado a ésta.

Por lo general, un SGBD proporciona los siguientes servicios:

- **Permite la definición de la base de datos mediante el lenguaje de definición de datos (DDL – Data Description Language).** Este lenguaje permite

especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos. Todo esto se almacenará en la base de datos.

- Permite la inserción, actualización, eliminación y consulta de datos mediante el lenguaje de manejo de datos (DML - Data Manipulation Language).
- Proporciona un acceso controlado a la base de datos mediante:
 - o un sistema de seguridad, por lo que los usuarios no autorizados no puedan acceder a la base de datos, mediante el lenguaje de control de datos (DCL - Lenguaje de Control de Datos);
 - o un sistema de integridad que mantiene la integridad y la consistencia de los datos;
 - o un sistema de control de concurrencia que permite el acceso compartido a la base de datos;
 - o un diccionario de datos o catálogo accesible por el usuario que contiene la descripción de los datos de la base de datos.

A diferencia de los sistemas de archivos, el SGBD gestiona la estructura física de los datos y su almacenamiento. Con esta funcionalidad, el SGBD se convierte en una herramienta de gran utilidad. Sin embargo, desde el punto de vista del usuario, se podría discutir que los SGBD han hecho las cosas más complicadas, ya que ahora los usuarios ven más datos de los que realmente quieren o necesitan, ya que ven la base de datos completa . Conscientes de este problema, los SGBD proporcionan un mecanismo de vistas mediante el lenguaje de definición de vistas (VDL – View Definition Language), que permite que cada usuario tenga su propia vista o visión de la base de datos. El lenguaje de definición de datos permite definir vistas como subconjuntos de la base de datos.

Las vistas, además de reducir la complejidad, permiten que cada usuario vea sólo la parte de la base de datos que necesita, tienen otras ventajas:

- Las vistas proporcionan un nivel de seguridad, ya que permiten excluir datos para que ciertos usuarios no los vean.
- Las vistas proporcionan un mecanismo para que los usuarios vean los datos en el formato que desean. •

Una vista representa una imagen consistente y permanente de la base de datos, incluso si la base de datos cambia su estructura.

NOTA: Hay dos tipos de lenguajes de manejo de datos: los procedurales y los no procedurales. Estos dos tipos se distinguen por la forma en que acceden a los datos. Los lenguajes procedurales manipulan la base de datos registro a registro, mientras que los no procedurales operan sobre conjuntos de registros. En los lenguajes procedurales se especifica qué operaciones deben realizarse para obtener los datos, mientras que en los lenguajes no procedurales se especifica qué datos deben obtenerse sin decir cómo hacerlo. El lenguaje no procedural más utilizado es el SQL que, de hecho, es un estándar y es el lenguaje de los SGBD relacionales. SQL asume el papel, por tanto, de DDL, DML, DCL y VDL.

Funciones de los SGBD

Los servicios o funciones que debe ofrecer todo SGBD son.

1. Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Ésta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).
2. Un SGBD debe proporcionar un catálogo en el que se almacenan las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es el que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos).

Normalmente, un diccionario de datos almacena:

- Nombre, tipos y tamaño de los datos.

- Nombre de las relaciones entre los datos.

Restricciones de integridad sobre los datos.

- Número de los usuarios autorizados a acceder a la base de datos.
- Esquemas externo, conceptual e interno y correspondencia entre los esquemas.

- Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.

Algunos de los beneficios que reporta el diccionario de datos son los siguientes:

La información sobre los datos se puede almacenar de forma centralizada. Esto ayuda a mantener el control sobre los datos, como un recurso que son. El significado de los datos se puede definir, lo que ayudará a los usuarios a entender el propósito de éstos. La comunicación se simplifica ya que se almacena el significado exacto.

El diccionario de datos también puede identificar al usuario o usuarios que poseen los datos o que los accedan.

Las redundancias y las inconsistencias se pueden identificar más fácilmente puesto que los datos están centralizados.

Se puede tener un historial de los cambios realizados sobre la base de datos.

El impacto que puede producir un cambio se puede determinar antes de que sea implementado, ya que el diccionario de datos mantiene información sobre cada tipo de dato, todas sus relaciones y todos sus usuarios. Se

puede hacer respetar la seguridad.

Se puede garantizar la integridad.

Se puede proporcionar información para auditorías.

3. Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos. Una transacción en el sistema informático de la empresa inmobiliaria sería dar de alta a un empleado o eliminar un inmueble. Una transacción algo más complicada sería eliminar a un empleado y reasignar sus inmuebles a otro empleado. En este caso es necesario realizar varios cambios sobre la base de datos. Si la transacción falla durante su realización, por ejemplo porque falla el hardware, la base de datos quedará en estado inconsistente.

Algunos de los cambios se habrán hecho y otros no, por tanto, los cambios realizados tendrán que ser deshechos para devolver la base de datos a un estado consistente.

4. Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es permitir que varios usuarios tengan acceso concurrente a los datos que comparten. El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está actualizando datos, pueden interferir de forma que se produzcan inconsistencias en la base de datos. El SGBD debe encargarse de que estas interferencias no se produzcan en el acceso simultáneo.
5. Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos debe devolverse a un estado consistente. Este fallo puede deberse a un fallo en algún dispositivo hardware o un error del software, que hagan que el SGBD aborte, o puede ser debido a que el usuario detecte un error durante la transacción y el aborto antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.

6. Un SGBD debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.

7. Un SGBD debe ser capaz de integrarse con algún software de comunicación.

Muchos usuarios acceden a la base de datos desde terminales. A veces estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD. En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD debe realizarse a través de una red. En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes y responde de forma similar. Todas estas transmisiones de mensajes son manejadas por el gestor de comunicaciones de datos.

8. Un SGBD debe proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realicen sobre estos datos, sigan unas ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otra forma de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar. Por ejemplo, puede establecerse la restricción de que cada empleado no puede tener asignados más de diez inmuebles. En este caso sería deseable que el SGBD controlara que no se sobrepase ese límite cada vez que se asigne un inmueble a un empleado.

9. Un SGBD permitirá que se mantenga la independencia entre los programas y la estructura de la base de datos. La independencia de datos se logra mediante las vistas o subesquemas. La independencia de datos física es más fácil de conseguir, de hecho existen varios tipos de cambios que se pueden dar sobre la estructura física de la base de datos sin afectar a las vistas. Sin embargo, conseguir una completa independencia de datos lógica es más difícil.

Agregar una nueva entidad, un atributo o una relación puede ser sencillo, pero no es tan sencillo eliminarlos.

10. Un SGBD debe proporcionar una serie de herramientas que permiten administrar la base de datos de forma efectiva. Algunas herramientas trabajan a nivel externo, por lo que habrán sido producidas por el administrador de la base de datos. Las herramientas que trabajan a nivel interno deben ser proporcionadas por el distribuidor del SGBD.

Algunas de ellas son:

- Herramientas para importar y exportar datos.

- Herramientas para monitorizar el uso y funcionamiento de la base de datos. •
- Programas de análisis estadístico para examinar las prestaciones o las estadísticas de utilización. •
- Herramientas para reorganización de índices.
- Herramientas para aprovechar el espacio dejado en el almacenamiento físico por los registros borrados y que consoliden el espacio liberado para reutilizarlo cuando sea necesario.

Ventajas

Los sistemas de bases de datos presentan numerosas ventajas como:

- Menor redundancia de datos.
 - Más información sobre los datos.
 - Mayor control sobre la consistencia de los datos.
 - Compartición de datos (conurrencia).
 - Mejora en la integridad de los datos.
 - Mejora en la seguridad -
- Mejora en la accesibilidad a los datos sin necesidad de ningún programa (SQL).
- Mejora en los servicios de copias de seguridad.

Inconvenientes

- Gran cantidad de espacio de disco y de memoria RAM para trabajar eficientemente.
- Coste económico elevado, tanto por el gestor como por el equipamiento adicional.
- Coste de la conversión, tanto de migración de datos como de enseñanza a la plantilla de su uso.
- Vulnerable a los fallos por el hecho de que todo esté centralizado.

Lenguajes de los SGBD

Los SGBD deben ofrecer lenguajes e interfaces apropiadas para cada tipo de usuario: administradores de la base de datos, diseñadores, programadores de aplicaciones y

usuarios finales.

Lenguaje de definición de datos (DDL)

Una vez finalizado el diseño de una base de datos y elegido un SGBD para su implementación, el primer paso consiste en especificar el esquema conceptual y esquema interno de la base de datos, y la correspondencia entre ambos. En muchos SGBD, para definir ambos esquemas, se utiliza el lenguaje de definición de datos

(DDL). El SGBD posee un compilador de DDL cuya función consiste en procesar las sentencias del lenguaje para identificar las descripciones de los distintos elementos de los esquemas y almacenar la descripción del esquema en el catálogo o diccionario de datos (metadatos).

El DDL modifica el ESQUEMA de la BD. Son instrucciones DDL: CREATE, ALTER o DROP.

Lenguaje de manejo de datos (DML)

Una vez creados los esquemas de la base de datos, los usuarios necesitan un lenguaje que les permita manipular los datos de la base de datos: realizar consultas, inserciones, eliminaciones y modificaciones. Este lenguaje es lo que se denomina lenguaje de manejo de datos (DML).

Hay dos tipos de DML: los procedurales y no procedurales. Con un DML procedural el usuario (normalmente será un programador) especifica qué datos se necesitan y cómo obtenerlos. Esto significa que el usuario debe especificar todas las operaciones de acceso a datos llamando a los procedimientos necesarios para obtener la información requerida. Estos lenguajes acceden a un registro, lo procesan y en base a los resultados obtenidos, acceden a otro registro, que también deben procesar. Así se va accediendo a registros y se van procesando hasta que se obtienen los datos deseados. Las sentencias de un DML procedural deben estar embebidas en un lenguaje de alto nivel, ya que se necesitan sus estructuras (bucles, condicionales, etc.) para obtener y procesar cada registro individual. A este lenguaje se le denomina lenguaje anfitrión. Las bases de datos jerárquicas y de red utilizan DML procedurales.

Un DML no procedural se puede utilizar de forma independiente para especificar operaciones complejas sobre la base de datos de forma concisa. Los DML no procedurales permiten especificar los datos a obtener en una consulta o los datos a actualizar, mediante una sola y sencilla sentencia. El usuario o programador especifica qué datos desea obtener sin decir cómo acceder a ellos.

SGBD traduce las sentencias del DML en uno o varios procedimientos que manipulan los conjuntos de registros necesarios. A los DML no procedurales también se les denomina declarativos. Las bases de datos relacionales utilizan DML no procedurales, como SQL (Structured Query Language) o QBE (Query-By-Example). Los lenguajes no procedurales son más fáciles de aprender y de usar que los procedurales, y el usuario debe hacer menos trabajo, siendo el SGBD quien hace la mayor parte.

El DML modifica el ESTADO de la BD. Son instrucciones DML: INSERT, UPDATE, DELETE o SELECT.

Lenguaje de control de datos (DCL)

Los lenguajes de control de datos contienen elementos útiles para trabajar en entornos multiusuario en los que son importantes la protección de datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como elementos para coordinar la compartición de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.

El DCL controla el acceso a los datos de la BD. Son instrucciones DCL: GRANT o REVOKE.

Clasificación de los sistemas gestores de bases de datos

Según el modelo lógico utilizado

El criterio principal que se utiliza para clasificar a los SGBD es el modelo lógico en el que se basan. Los modelos lógicos empleados con mayor frecuencia en los SGBD comerciales actuales son el relacional, el de red y el jerárquico. Algunos SGBD más modernos se basan en modelos orientados a objetos.

El modelo relacional se basa en el concepto matemático denominado "relación", que gráficamente se puede representar como una tabla. En el modelo relacional, los datos y las relaciones existentes entre los datos se representan mediante estas relaciones matemáticas, cada una cuyo nombre es único y con un conjunto de columnas.

En el modelo relacional la base de datos es percibida por el usuario como un conjunto de tablas. Esta percepción es sólo a nivel lógico (en los niveles externo y conceptual de la arquitectura de tres niveles), puesto que a nivel físico puede estar implementada mediante diferentes estructuras de almacenamiento. Oracle y MySQL, por ejemplo, son SGBD relacionales muy populares. Es el modelo que vamos a estudiar durante este curso.

En el modelo de red los datos se representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos, que son líderes en la implementación física. Los registros se organizan como un grafo: los registros son los nodos y los arcos son los conjuntos. El SGBD de red más popular es el sistema IDMS.

El modelo jerárquico es un tipo de modelo de red con algunas restricciones. De nuevo, los datos se representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos. Sin embargo, en el modelo jerárquico cada nodo puede tener un solo padre. Una base de datos jerárquica puede representarse mediante árbol: los registros son los nodos, también denominados segmentos, y los arcos son los conjuntos. El SGBD jerárquico más importante es el sistema IMS.

La mayoría de los actuales SGBD comerciales están basados en el modelo relacional, mientras que los sistemas más antiguos estaban basados en el modelo de red o el modelo jerárquico. Estos dos últimos modelos requieren que el usuario tenga conocimiento de la estructura física de la base de datos a la que se accede, mientras que el modelo relacional proporciona mayor independencia de datos. Se dice que el modelo relacional es declarativo (se especifica qué datos se deben obtener) y los modelos de red y jerárquico son navegacionales (se especifica cómo deben obtenerse los datos).

El modelo orientado a objetos define una base de datos en términos de objetos, sus propiedades y operaciones. Los objetos con la misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías o grafos acíclicos. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados métodos. Algunos SGBD relacionales existentes en el mercado han estado extendiendo sus modelos para incorporar conceptos orientados a objetos. A estos SGBD se les conoce como sistemas objeto-relacionales.

Según el número de usuarios a los que presta servicio

Un segundo criterio para clasificar a los SGBD es el número de usuarios a los que da servicio el sistema. Los sistemas monousuario sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales. Los sistemas multiusuario, entre los que se encuentran la mayor parte de los SGBD, atienden a varios usuarios al mismo tiempo.

Según la distribución física de la base de datos

Un tercer criterio es el número de sitios en los que está distribuida la base de datos. Casi todos los SGBD son centralizados: sus datos se almacenan en un solo computador. Los SGBD centralizados pueden atender a varios usuarios, pero el SGBD y la base de datos en sí residen por completo en una sola máquina. En los SGBD distribuidos la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red.

Los SGBD distribuidos homogéneos utilizan el mismo SGBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a diversas bases de datos autónomas preexistentes almacenadas en SGBD distribuidos heterogéneos. Esto da lugar a los SGBD federados o sistemas multibase de datos en los que los SGBD participantes tienen cierto grado de autonomía local. Muchos SGBD distribuidos utilizan una arquitectura cliente-servidor.